

# Generic Opinion Mining System for Decision Support

Dr.P.G.Naik <sup>#1</sup>, S.S.Jamsandekar <sup>\*2</sup>, K.S.Mahajan <sup>#3</sup>

<sup>#</sup> Department of Computer Studies, CSIBER,  
Kolhapur, Maharashtra, India

<sup>1</sup> [pgnaik@siberindia.edu.in](mailto:pgnaik@siberindia.edu.in)

<sup>2</sup> [ssjamsandekar@siberindia.edu.in](mailto:ssjamsandekar@siberindia.edu.in)

<sup>3</sup> [ksmahajan@siberindia.edu.in](mailto:ksmahajan@siberindia.edu.in)

**Abstract**— Social networking sites prove to be indispensable tools for decision making owing to the large repository of user views accumulated over a period of time. Such a real data can be exploited for various purposes such as making buying decisions, analysing the user views about new product launched by a company, product promotion campaign, impact of policy decisions made by a political party on society etc. In the current work the authors have proposed a generic model for feature based polarity determination by sentiment analysis of tweets. This model has been implemented by the seamless integration of R tool, XML, JAVA, Link Parser A practical multistep system, in place, efficiently extracts data from tweet text, pre-process the raw data to remove noise, and tags their polarity. Data used in the current study is derived from online product feature based reviews collected from tweeter tweets. Link parser version 4.1 b is employed for parsing a natural sentence which is broken into multiple tokens corresponding to noun and adjective before being stored in a persistent storage medium. The objectivity score is determined using SentiWordNet 3.0 lexical resource which is parsed using a tool implemented in Java. The linguistic hedges are taken care of using Zadeh's proposition which modifies the final objectivity score. The objectivity score so computed, provides the necessary guidelines in influencing decisions. The authors have tested the model for product purchase decisions of two different sets of products, smart phone and laptop based on predefined set of features. The model is generic and can be applied to any set of products evaluated on a predefined set of features.

**Keyword**- Feature Extraction, Fuzzy Hedges, Linguistic Variables, Link Parser, Natural Language Processing, R Software, SentiWordNet, Twitter.

## I. INTRODUCTION

Sentiment is not all about mere facts and figures but incorporates feelings, emotions, opinions or attitudes of an individual towards the situation or a topic [1]. Across the globe users share their opinions on variety of topics and discuss several current issues on many social networking sites and blogs such as Twitter, face book etc., making them valuable platforms for tracking and analyzing public sentiment. Over past few years Twitter has gained tremendous popularity among social media for expressing user views or opinions on their topic of interest and these timely tweets from users restrain sentiments. Analysis of such sentiments can provide an economical and effective way to expose timely public sentiment, which is critical for decision making in various domains. Insight into user views and applications from sentiment analysis have been useful in hot areas like politics, policy making, psychology, business intelligence, customer buying behavior etc. Focusing on business intelligence and understanding customers' opinion and subjectivity is regarded as an important task in various domains (e.g., marketing). Customers post reviews of purchased products on a continual basis, which results in incremental accumulation of a lot of reviews written in natural language and such opinions if propagated to other users and might make a significant influence on them in making purchase decisions, for example.

In this paper, the authors propose a model for feature based polarity determination by sentiment analysis of tweets. This model has been implemented by the seamless integration of R tool, XML, JAVA, Link Parser. Thereby, a practical multistep system, has been implemented to efficiently collect data, preprocess to remove noise, extract product and its features from the tweet text, and tagging their polarity. Data used in the current study is derived from online product feature based reviews collected from tweeter tweets.

The rest of the paper is organized as follows. Section(II) describes the prior related work in opinion mining, followed by Section (III), Section (IV), Section (V) describing tools and techniques employed, design of an

algorithm for computation of objectivity score, and the proposed methodology in construction of the model, respectively. Section(VI) describes the implementation of proposed system. Section (VII) discusses the experimental evaluation. Finally, conclusion and scope for the future work is discussed in Section (VIII).

## II. PRIOR RELATED WORK:

In literature, there exist quite a fistful of papers on opinion mining of social networking data. However, all these papers focus on a single domain. Duc Nguyen Trung and Jason J. Jung [2] have designed a mathematical model of information propagation. Their model is based on fuzzy propagation for opinion mining by sentiment analysis of online social networks. Wu et. al [3] have employed in their work, phrase dependency parsing for opinion mining. In dependency grammar, structure is determined by the relation between a head and its dependents. The dependent is a modifier or complement and the head plays a more important role in determining the behaviours of the pair. Authors in [4] have proposed an opinion mining system to classify products based on feature incorporated with fuzzy hedges to emulate the effect of linguistic modifiers, concentrators, and dilators. Xing Fang and Justin Zhan [5] have modeled in their work, the fundamental problem of sentiment analysis, sentiment polarity categorization for both sentence-level categorization and review-level categorization of online product reviews. They have employed in their work, Naïve Bayesian, Random Forest, and Support Vector Machine classification models for categorization. In their Opinion Mining and Sentiment analysis paper Md. Daiyan et al. [6] have presented reviews on different methods, tools and dataset used for sentiment analysis such as, Document Level Sentiment Classification, Classification Based on Supervised Learning, Classification Based on Unsupervised Learning, Sentence Level Sentiment Classification, Mining Comparative and Superlative Sentences to name few of them. According to Dave et al. [7] an ideal opinion-mining tool would be to process a set of search results for a given item, generating a list of product attributes (quality features, etc.) and aggregating opinions about each of them (poor, mixed, good). Pang et al. [8] have employed three Machine learning methods such as Naïve Bayesian, Support Vector Machine and maximum entropy classification where the authors have taken internet movies keywords as a dataset. The authors have come up with a model wherein rating were automatically extracted with three categorizations, positive, negative and neutral. In their work the authors have employed standard Bag of feature framework. According to the authors' findings, in terms of relative performances Naïve Bayesian do the worst and support vector machine results are best, however differences are not very large. Bo pang et al. [9] have applied Meta algorithm, which is based on metric labeling for the formulation of the problem. The authors have considered generalizing their model to finer grained scales and attempt to apply numerical rating such as three stars or four stars. The authors have presented three types of algorithms, Ones Vs All (OVA), regression and metric labeling which can be distinguished by how they can explicitly leverage similarity between items and between labels. They have presented in their results the type of item similarity measure to apply based on the positive sentences. The authors have considered Movie rating as dataset to evaluate pairs of reviews in order to find out review of each pair and classify them into more positive, less positive and, a positive category. There are three class of metric labeling on top of OVA and regression which reveal that employing explicit similarities always improves the result often to a significant degree and yields the overall best accuracies.

## III. PROPOSED MODEL FOR OPINION MINING

In this section authors discuss the design of proposed generic opinion mining system for product comparison based on linguistic variables with the presence or absence of linguistic fuzzy hedges. In the current work, authors have performed opinion mining on a dataset of online user reviews collected using a Twitter application. The architecture for the current system is depicted in Figure 1.

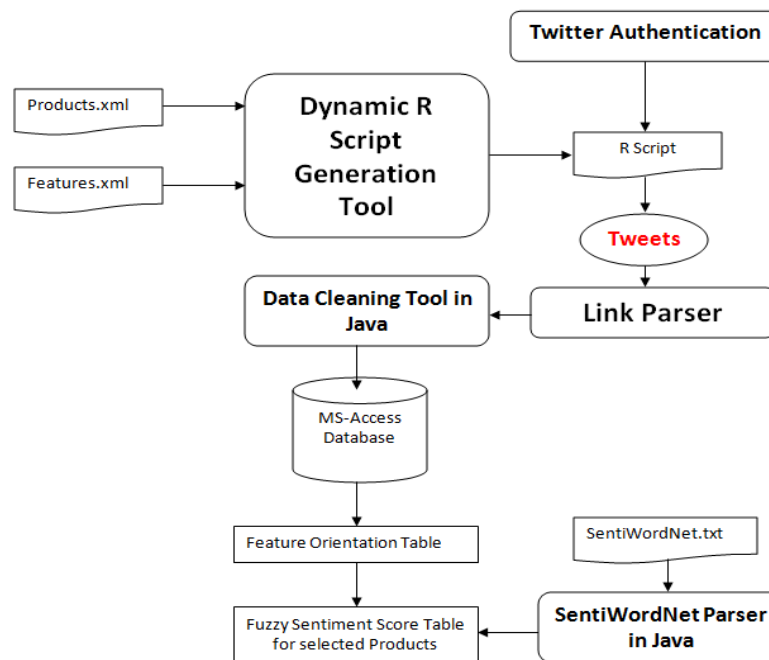


Figure 1. Architecture of Generic Opinion Mining System

The opinion mining system begins with the acceptance of input from an end user pertaining to the products to be compared on a pre-defined feature set which is stored in a persistent storage medium in XML format for future use. The Dynamic R Script Generation tool captures this information, concatenates the product with the feature information and dynamically generates the R Script for extracting tweets by connecting to twitter application. The dynamic R Script generated is listed in Appendix A. The raw data extracted from Twitter application contains unstructured user views. Such a raw data often contains substantial noise pertaining to errors in spelling, grammar, punctuation, extensive use of acronyms and re-tweets which needs to be cleaned for removing noise before classifying them based on their sentiment. Further, the feature may be modulated with concentrating, dilating or modifying fuzzy hedges which should be properly accounted for before generating the objectivity score. The cleaned tweets are fed to link grammar parser which extracts the tokens pertaining to noun, adjective describing the noun and verb. The linguistic hedges are extracted separately and stored in a database along with the tokens. The mined feature set is tabulated in a feature oriented table.

#### IV. TOOLS AND TECHNIQUES

##### A. *R Software*

R is an integrated suite of software applications commonly employed for data manipulation, calculation and graphical display.

R has

- an effective data handling and storage capability,
- a large, coherent, integrated collection of intermediate tools focusing on data analysis
- graphical libraries for data analysis and display and
- a well developed, simple and effective programming language (called ‘S’) which includes conditionals, loops, user defined recursive functions and input and output facilities. Most of the system supplied functions are themselves written in the S language.

The basic R libraries employed in the current work for extracting tweets are depicted in Table I. Each library involves multiple dependencies on other R libraries.

Table I. R Library Employed in Model Implementation

TwitteR RCurl RJSONIO stringr xlsx
--

B. *SentiWordNet3.0*

SentiwordNet 1.0 (Esuli and Sebastiani, 2006) was publicly made available for research Purposes. SentiWordNet 3.0, a lexical resource explicitly devised for supporting sentiment classification and opinion mining applications. It is the result of automatically annotating all WordNet synsets according to their degrees of positivity, negativity, and neutrality. SentiwordNet 1.0 and 3.0 differ (a) in the versions of WordNet which they annotate (WordNet 2.0 and 3.0, respectively), (b) in the algorithm used for automatically annotating WORDNET, which now includes (additionally to the previous semi-supervised learning step) a random-walk step for refining the scores [10]

Structure of SentiWordNet3.0

SentiWordNet 3.0 has a tabular structure as depicted in Figure 2. consisting of 6 columns corresponding to Pos , which can be one of a(adjective), n(noun), or v(verb), ID, positive score, negative score, synset terms with different weightage and glossary. The objectivity score is calculated as follows:

$$ObjScore = 1 - (PosScore + NegScore)$$

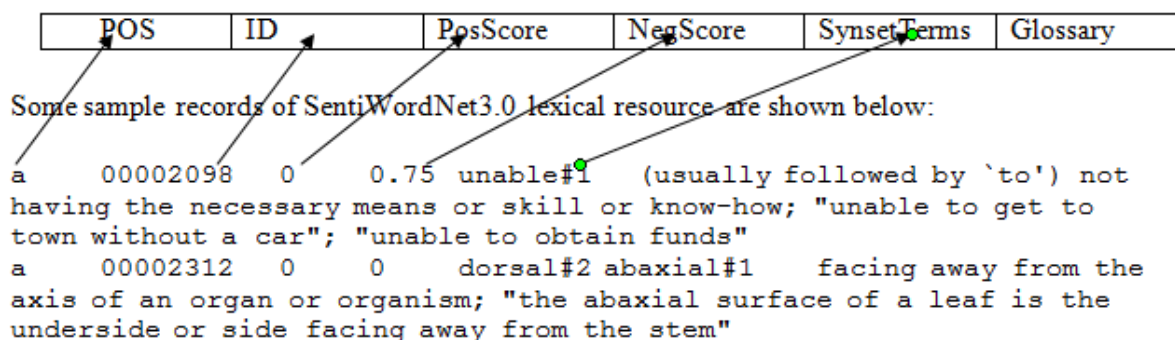


Figure 2. Structure of SentiWordNet 3.0

In the current work, the authors have employed SentiWordNet 3.0 for determination of polarity of synset terms.

C. *Link Grammar Parser*

Link grammar is closer to dependency grammar with two basic parameters: directionality and distance, but dependency grammar incorporates a head-dependent relationship which at the same time lacks directionality in the relations between words. Link grammar (LG) is a theory of syntax coined by Davy Temperley and Daniel Sleator which builds relations between pairs of words in contrast to constructing constituents in a tree-like hierarchy [11].

Colored Multiplanar Link Grammar (CMLG) is an extension of LG which allows crossing relations between pairs of words. For example, in a subject-verb-object language like English, the verb would look left to form a subject link, and right to form an object link. Nouns would look right to complete the subject link, or left to complete the object link. In a subject-object-verb language like Persian, the verb would look left to

form an object link, and a more distant left to form a subject link. Nouns would look to the right for both subject and object links.

### Syntax

Rightward links are represented as a +, and leftward links with a -. Optional links are contained in curly brackets {...}. Undesirable links are contained in any number of square brackets [...]. Multiple links are joined either by a conjunction & or a disjunction or. Each rule ends with a semicolon ;.

A basic rule file for an SVO language might look like:

```
<determiner>: D+;
<noun-subject>: {D-} & S+;
<noun-object>: {D-} & O-;
<verb>: S- & {O+};
```

Conversely, a rule file for a null subject SOV language might consist of the following links:

```
<noun-subject>: S+;
<noun-object>: O+;
<verb>: {O-} & {S-};
```

In the current work we have employed the link parser, version 4.1b which is being released under a GPL compatible license, making it freely available for research and commercial use [12].

### D. Twitter Authentication

Since the release of version 1.1 of Twitter API, OAuth handshake is made necessary for every user request which requires the verification of app at Twitter. The preliminary steps involved in Twitter authentication are enumerated below:

- Visit Twitter home page at to <https://dev.twitter.com/> and log in to your Twitter Account.
- Click on the drop-down menu in the upper right corner and click on “My Applications”
- Select “Create New Application” from the drop-down menu.
- Twitter requires a valid URL for the website. You can just type in <http://test.de>, if you do not need it any more. Leave the callback URL blank as shown in Figure 3.
- Click on “Create” which will redirect you to a screen with the entire OAuth setting of your new App. Leave the window open in the background.
- Continue to R and enter the following lines.

**Application Details**

Name: \*  
SentimentTEST  
Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description: \*  
Sentiment Analysis for Twitter  
Your application descriptor, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

Website: \*  
http://test.de/  
Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens. (If you don't have a URL yet, just put a placeholder here but remember to change it later.)

Callback URL:  
Where should we return after successfully authenticating? For @Anywhere applications, only the domain specified in the callback will be used. OAuth 1.0a applications should explicitly specify their oauth\_callback URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank.

Figure 3. Twitter Authentication.

```

reqURL <- "https://api.twitter.com/oauth/request_token"
accessURL <- "http://api.twitter.com/oauth/access_token"
authURL <- "http://api.twitter.com/oauth/authorize"
consumerKey <- "yourconsumerkey"
consumerSecret <- "yourconsumersecret"
twitCred<-
OAuthFactory$new(consumerKey=consumerKey,consumerSecret=consumerSecret,requestURL=reqURL,accessURL=accessURL,authURL=authURL)
download.file(url="http://curl.haxx.se/ca/cacert.pem", destfile="cacert.pem")
twitCred$handshake(cainfo="cacert.pem")
registerTwitterOAuth(twitCred)

```

- Replace “yourconsumerkey” and “yourconsumersecret” with the data provided by Twitter on your App page which is still open in the background.

### E. Linguistic Hedges

The tweets collected from the user views are often in the form of natural language sentences containing linguistic terms often modulated by fuzzy hedges either concentrating or dilating the intensity of terms and modifying the degree of membership. In the current work, we employ the model proposed by Zadeh [13] for updating the degree of membership corresponding to the fuzzy hedge. The effect of fuzzy hedges on membership functions is depicted in Figure 4 and Figure 5, for concentrator and dilator, respectively, for different values of  $\delta$ .

If the descriptor has a preceding hedge, its modified fuzzy score is calculated using

$$(\mu(s)) = 1 - (1 - (s))^\delta$$

Similar to Zadeh’s proposition, if the hedge is a concentrator, we choose  $\delta=2$  which gives us modified fuzzy concentrator score as indicated in Figure 3, while if the hedge is a dilator we choose  $\delta = \frac{1}{2}$  which gives us modified fuzzy dilator score as indicated in Figure 4.

$$f_c((s)) = 1 - (1 - (s))^2$$

$$f_d((s)) = 1 - (1 - (s))^{\frac{1}{2}}$$

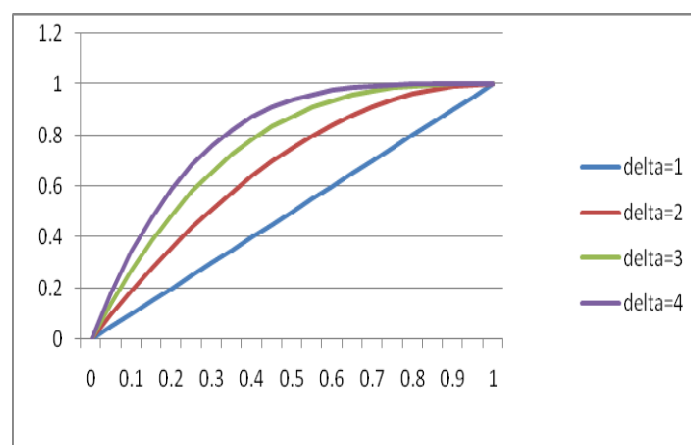


Figure 4. Effect of Concentrator on Membership Function.

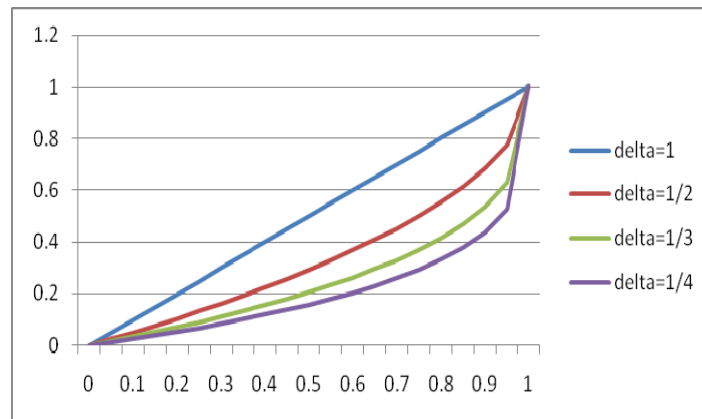


Figure 5. Effect of Dilator on Membership Function.

## V. ALGORITHM FOR COMPUTATION OF OBJECTIVITY SCORE

This section presents the design of an algorithm for computation of objectivity score for synset terms extracted from tweets by employing SentiWordNet3.0 lexical resource.

```

/*Data Structures for Storing synset terms and score */
struct Map
{
    char synsetterm;
    int rank;
    float score;
};
struct Dictionary
{
    char sysetterm[10];
    float objectivityscore;
};

/* Global Variable Declarartion */
char data[10][10];
float score;
char term[10][10];
int rank;
char s[10];
Map m[100];
Dictionary d[100];
float sum=0.0;
float dscore=0.0;
int count=0;

function computeObjectivityScore()
{
while(line=readLine("SentiWordNet.txt"))
{
    /* Skip Comments */
    if (!line[0]=="#")
    {
        data=line.split("\t");
        /*
            data[0]=WordTypeToken, which can be 'a', 'n' or 'v'
            data[1]=ID
            data[2]=Positive Score
        */
    }
}
}

```

```

    data[3]=Negative Score
    data[4]=Synset Term
    data[5]=Glossary of Terms
*/
/* Compute Score */
score = float(data[2])-float(data[3]);
terms=data[4].split(" ");
for (s in terms)
{
    termAndRank=s.split("#");
    rank=int(termAndRank[1]);
    m[count].synsetterm=s;
    m[count].rank=rank;
    m[count].score=score;

    /* Calculate weighted average. Weigh the synsets according to
    their rank.
    Score= 1/1*first + 1/2*second + 1/3*third ..... etc.
    Sum = 1/1 + 1/2 + 1/3 ...
    */

    dscore=score/rank;
    sum=sum+1/rank;
    objectivityscore= dscore/sum;
    d[count].synsetterm=s+"#+data[0];
    d[count++].score=objectivityscore;
}
}
}
}
function extract(char[10] word, char[10] type)
{
    char search_string=word + "#" + type
    for i= 1 to count
    {
        if (d[i].synsetterm == search_string)
            return d.objectivityscore;
    }
    return 0;
}

```

The working of the algorithm is indicated for a sample Score Calculation for synset term `a#good` and is depicted in Figure 6 along with the various formulae employed in the calculation.



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Token Type	ID	Pos Score	Neg Score		Rank	Score	Score/Rank	Sum				
2	a	1123148	0.75	0	good#1	1	0.75	0.75	1		<b>Formulae Used :</b> $G2 \rightarrow C2-D2$ $H2 \rightarrow (1/(F2))^*G2$ $I2 \rightarrow 1/F2$		
3	a	106020	0	0	good#2	2	0	0	0.5				
4	a	1129977	1	0	good#3	3	1	0.333333333	0.333333333				
5	a	1983162	1	0	good#4	4	1	0.25	0.25				
6	a	64787	0.625	0	good#5	5	0.625	0.125	0.2				
7	a	1586752	1	0	good#6	6	1	0.166666667	0.166666667				
8	a	2036934	0.75	0	good#7	7	0.75	0.107142857	0.142857143				
9	a	2226162	0.625	0	good#8	8	0.625	0.078125	0.125				
10	a	523364	0.625	0	good#9	9	0.625	0.069444444	0.111111111				
11	a	452883	0.5	0	good#10	10	0.5	0.05	0.1				
12	a	2273643	0.5	0	good#11	11	0.5	0.045454545	0.090909091				
13	a	1661289	0.375	0	good#12	12	0.375	0.03125	0.083333333				
14	a	1048762	0.625	0	good#13	13	0.625	0.048076923	0.076923077				
15	a	832784	0	0	good#14	14	0	0	0.071428571				
16	a	1808329	0.625	0	good#15	15	0.625	0.041666667	0.066666667				
17	a	1333477	0.75	0	good#16	16	0.75	0.046875	0.0625				
18	a	1171213	0.75	0	good#17	17	0.75	0.044117647	0.058823529				
19	a	1166413	0.875	0	good#18	18	0.875	0.048611111	0.055555556				
20	a	1116026	0.5	0	good#19	19	0.5	0.026315789	0.052631579				
21	a	1068306	0.375	0.125	good#20	20	0.25	0.0125	0.05				
22	a	775611	0.75	0	good#21	21	0.75	0.035714286	0.047619048				
23								2.31029427	3.645358705				
24								<b>Score</b>	<b>0.6338</b>				
25													

Figure 6. Computation of Objectivity Score for#good

### VI. MODEL IMPLEMENTATION

The implementation of the model is divided into the following phases.

**Phase 1 :** Twitter Authentication Phase

**Phase 2 :** Dynamic R Script Generation by parsing XML configuration files.

**Phase 3 :** Data cleaning to remove re-tweets, errors in spelling, grammar, punctuation, etc.

**Phase 4 :** Splitting the tweets into different tokens using link grammar parser.

**Phase 5 :** Data cleaning and storing adjectives and nouns in a persistent database.

**Phase 6 :** Generation of feature orientation table.

**Phase 7 :** Determining the objectivity score of each synset specified in the linguistic feature set and applying Zadeh’s equation to modify the score for any fuzzy hedges, if present. To facilitate this SentoWordNet dictionary is parsed in Java and cumulative objectivity score is determined.

**Phase 8 :** Graphical display of objective scores of feature sets for specified products.

The model is generic and can be applied for the comparison of any products based on the feature set specified by an end user. The product and feature details are stored in XML configuration files which is parsed using MS-XML parser.

Interaction between the various modules is depicted in Figure 7.

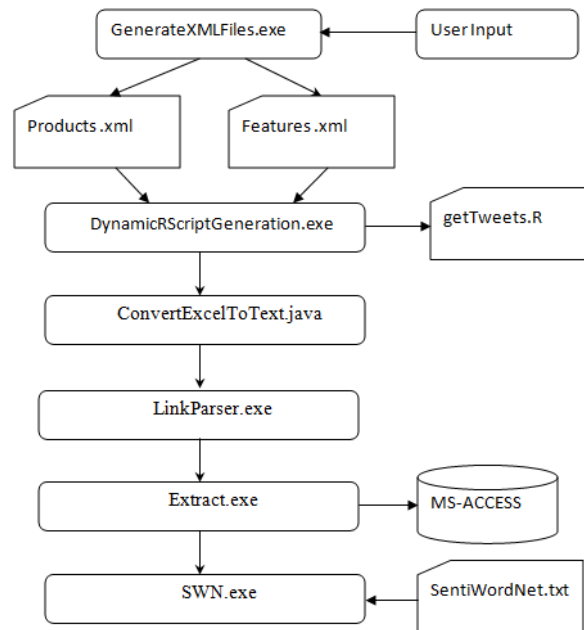


Figure 7. Interaction Between Various Modules of the Application.

```

GenerateXMLFiles.exe
DynamicRScriptGeneration.exe
set path=%path%;C:\Program Files\R\R-3.2.3\bin
rscript tweets.r
Set Path=%path%;C:\Program Files\Java\JDK1.7\bin
javac ConvertExcelToText.java
java ConvertExcelToText
LinkParser < input.txt > output.txt
javac Extract.java
java Extract
javac SWN.java
java SWN SentiWordNet.txt
pause
  
```

Figure 8. Contents of run.bat

## VII RESULTS AND ANALYSIS

The model presented above is implemented in multi-tier architecture. The Graphical User Interface in the presentation tier for accepting product and feature related information from an end user is depicted in Figure 9. The data entered by the user is stored persistently in XML file format.



Figure 9. Graphical User Interface for User Interaction

The corresponding XML files, Products.xml and Features.xml generated are show in Figure 10. along with the Document Type Definition for each.

```

Products.xml
<products>
  <product>
<name>Laptop1</name>
  </product>
  <product>
<name>Laptop2</name>
  </product>
  <product>
<name>Laptop3</name>
  </product>
</products>
    
```

```

DTD
<?XML Verion="1.0">
<!ELEMENT products(product+)>
  <!ELEMENT product(name)>
  <!ELEMENT name(#PCDATA)>
    
```

Features.xml	DTD
<features>	<?XML Verion="1.0">
<feature>	<!ELEMENT features(feature+)>
<name>Battery Life</name>	<!ELEMENT feature(name)>
</feature>	<!ELEMENT name(#PCDATA)>
<feature>	
<name>Display</name>	
</feature>	
<feature>	
<name>Drive/RAM</name>	
</feature>	
<feature>	
<name> Keyboard/Touch Pad </name>	
</feature>	
<feature>	
<name>Portability</name>	
</feature>	
<feature>	
<name> Processor</name>	
</feature>	
</features>	

Figure 10. Auto Generated XML Files Storing Product and Feature Related Information

Figure 11. shows parsing cleaned twitter data using link parser for extracting noun and adjective.

```

C:\ F:\sentiment\linkparser\link41b.exe
Opening ./data/words/words.adv.1
Opening ./data/words/words.adv.2
Opening ./data/4.0.knowledge
Opening ./data/4.0.constituent-knowledge
Opening ./data/4.0.affix
linkparser> battery is excellent
+++Time                               0.00 seconds (0.00 total)
Found 1 linkage (1 had no P.P. violations)
Unique linkage, cost vector = <UNUSED=0 DIS=0 AND=0 LEN=3>
  +---Ss--+---Pa--+
  |         |         |
battery.n is.v excellent.a
linkparser>
    
```

Figure 11. Parsing Natural Language Sentence using Link Parser

Figure 12. shows calculating objectivity score for fuzzy linguistic variables using SentiWordNet parser in Java and SentiWordNet dictionary.

```

C:\> Select C:\WINDOWS\system32\cmd.exe
C:\Program Files\Java\jdk1.5.0\bin>java SWN SentiWordNet.txt
poor #a -0.48469387755102045
fair #a 0.2713555073838233
good #a 0.633763219823854
excellent #a 1.0
C:\Program Files\Java\jdk1.5.0\bin>
    
```

Figure 12. computation of Objectivity Score for Fuzzy Linguistic Variables using SentiWordNet 3.0

The model developed in Section III is applied for the comparison of two products based on the feature set specified by an end user. Table 2. depicts fuzzy score for SmartPhone1 and SmartPhone2 pertaining to three features, display, battery life and camera.

Table 2. Fuzzy Score for Selected Smart Phones

	Display	BatteryLife	Camera
SmartPhone1	17.64	42.10767	30.23
SmartPhone2	8.925	39.96622	40.11

The relative comparison between different features of two smart phones is depicted in Figure 13.

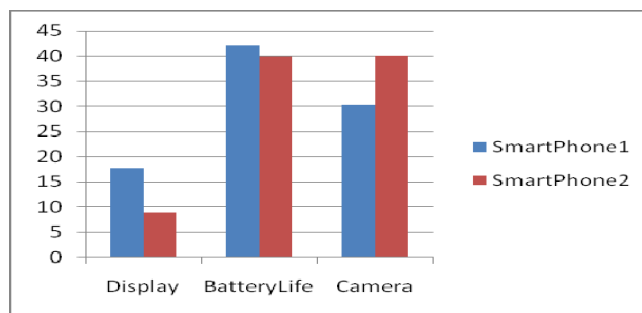


Figure 13. Feature-wise Comparison of Selected Smart Phones.

It is observed from Figure that SmartPhone2 has better score for camera whereas SmartPhone1 out scores in display and battery life. The same model is employed for comparison of three laptops based on five features depicted in Table 3.

Table 3. Fuzzy Score for Selected Laptops.

	Battery Life	Display	Hard Drive/RAM	Keyboard/TouchPad	Portability	Processor
Laptop1	50.7	60.7	65.7	23.5	50.5	70.4
Laptop2	30.5	55.6	57.8	15.4	60.5	56.8
Laptop3	40.4	40.9	30.7	20.5	70.4	34.7

Figure 14. shows the relative comparison of three laptops based on specified features.

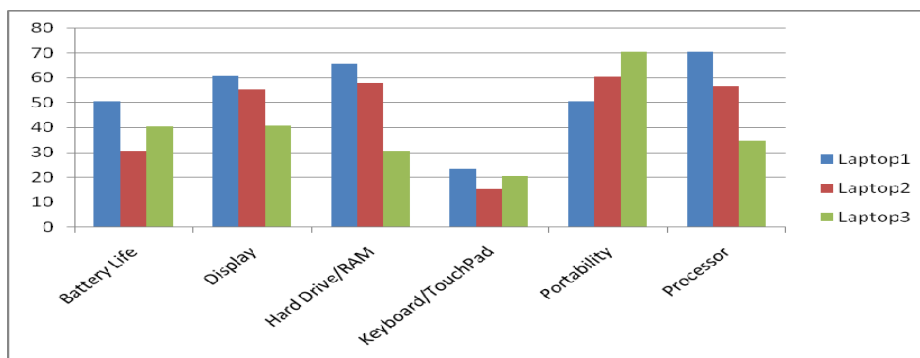


Figure 14. Feature-wise Comparison of Selected Laptops

As is revealed from the Figure 14 Laptop1 is better compared to other laptops in processor speed, battery life, display, keyboard and memory, whereas laptop3 has better portability compared to its counter parts based on the users reviews collected from Tweeter. Hence Laptop1 is recommended if portability is not an issue. The model can be used for comparison of any brand of products on specified feature set which proves as an indispensable tool in production selection based on user feedback.

### VIII. CONCLUSION AND SCOPE FOR FUTURE WORK.

Social networking sites lay a platform for the users for sharing their unbiased opinions on variety of topics of interest, products etc. and for discussing several current issues on a continual basis. Such activities result into large repository of unstructured user views which need to be cleaned and mined before extracting any useful information from it. In the current work the authors have proposed a generic model for feature based polarity determination by sentiment analysis of tweets. This model has been implemented by the seamless integration of R tool, XML, JAVA, and Link Parser. The objectivity score is determined using SentiWordNet 3.0 lexical resource. The linguistic hedges are taken care of using Zadeh's proposition which modifies the final objectivity score. The objectivity score so computed, provides the necessary guidelines in influencing decisions. The model is successfully tested for two distinct products, smart phone which are evaluated based on a set of features corresponding to display, battery life and camera and a set of laptops which are evaluated on batter life, display, hard drive/RAM, keyboard/touch pad, portability and processor. The model is generic and can be applied for influencing the purchase decisions of any set of products based on real-time user views.

### .REFERENCES:

- [1] Chandar. M. Jadhav, Ashwini P. Pati," Analyzing Variation of Public Sentiment on Twitter", International Journal of Advanced Research in Computer Science and Software Engineering Volume 4, Issue 11, November 2014.
- [2] Duc Nguyen Trung and Jason J. Jung, "Sentiment Analysis Based on Fuzzy Propagation in Online Social Networks: a Case study on TweetScope", Computer Science and Information Systems 11(1)
- [3] Yuanbin Wu, Qi Zhang, Xuanjing Huang, Lide Wu, "Phrase Dependency Parsing for Opinion Mining", EMNLP '09 Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, 2009, Volume 3.
- [4] Mita K. Dalal and Mukesh A. Zaveri, "Opinion Mining from Online User Reviews Using Fuzzy Linguistic Hedges", Hindawi Publishing Corporation Applied Computational Intelligence and Soft Computing Volume 2014.
- [5] Xing Fang and Justin Zhan, "Sentiment analysis using product review data", Journal of Big Data, Springer Open Journal, Volume 2 (5)2015.
- [6] Md. Daiyan, Dr. S. K. Tiwari, Manish Kumar, M. Aftab Alam, "A Literature Review on Opinion Mining and Sentiment Analysis", International Journal of Emerging Technology and Advanced Engineering, Volume 5, Issue 4, April 2015.
- [7] D. Dave, A. Lawrence, and D. Pennockm, "Mining the Peanut Gallery - Opinion Extraction and Semantic Classification of Product Reviews", Proceedings of International World Wide Web Conference (WWW'03), Hungary, 2003, pp. 519 - 528.
- [8] Pang, B., L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment classification using machine learning techniques", EMNLP '02 Proceedings of the ACL-02 conference on Empirical methods in natural language processing - ACM New York, Volume 10, pp. 79-86
- [9] A. Mukherjee, B. Liu, N. Glance, "Spotting Fake Reviewer Groups in Consumer Reviews", WWW '12 Proceedings of the 21st international conference on World Wide Web- ACM, pp.191-200
- [10] Andrea Esuli and Fabrizio Sebastiani. 2006. SENTIWORDNET:A publicly available lexical resource for opinion mining. In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC'06), pages417-422, Genova, IT.
- [11] <http://www.abisource.com/projects/link-grammar/>
- [12] <http://www.link.cs.cmu.edu/link>
- [13] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning-II," Information Sciences, vol.8,no.4,part3,pp.301-357,1975.

## APPENDIX A

```

Code for Parsing XML file using Microsoft XML Parser
fname = App.Path & "\products.xml"
fname = App.Path & "\features.xml"
Set objXML = New MSXML2.DOMDocument
If Not objXML.Load(fname) Then
    Err.Raise objXML.parseError.errorCode, , objXML.parseError.reason
End If
Set objElem = objXML.selectSingleNode("features")
' iterate its sub-nodes
cntFeatures = 0
For Each objSub In objXML.selectNodes("//feature")
    Set objElem = objSub.selectSingleNode("name")
    cntFeatures = cntFeatures + 1
    features(cntFeatures) = objElem.Text
Next

For i = 1 To cntProducts
    For j = 1 To cntFeatures
        prodfeature = products(i) & " " & features(j)
        GenerateRScript(prodFeature)
    Next j
Next i

```

**Dynamically Generated R Script****getTweets.R****# Install and Activate Packages**

```

library(twitterR)
library(RCurl)
library(RJSONIO)
library(stringr)
library(xlsx)

```

**# Create Twitter Connection**

```

setup_twitter_oauth('gOBUXia646ut7he3t8MIRA9Wx','UncYacZPXk3hjuHJjugnHiaqLZJKXrA9avKXv3UC
MmILe8qmTX','4853713368-
Gd1i5OCQt7FwuLaVodzX5nPrP0lmsA5qLybeHMU','bpNsfm8Rh3ry8RwgHXJN2heNDDg15q52yqKJnmap
nyCl')

```

**# harvest some tweets**

```

some_tweets = searchTwitter("Smartphone1 camera ", n=3000, lang="en", since="2014-08-20")
tweet.df <- twListToDF(some_tweets)

```

**# Transform tweets list into a data frame**

```

Created <- tweet.df$created
counts <- table(as.Date(Created))[-1]
dates <- as.Date(names(counts))
write.xlsx(x = tweet.df, file = "test.excelfile.xlsx", sheetName = "TestSheet", row.names = TRUE)
a<-read.xlsx(file = "test.excelfile.xlsx",sheetName = "TestSheet")
table(a$text)
plot(dates, counts, type="h")
some_txt = sapply(some_tweets, function(x) x$get_text())
some_txt = gsub("(RT|via)((?:\b\\W*@\w+)+)", "", some_txt)

```

**# remove at people**

```

some_txt = gsub("@\w+", "", some_txt)

```

```

# remove punctuation
some_txt = gsub("[[:punct:]]", "", some_txt)
# remove numbers
some_txt = gsub("[[:digit:]]", "", some_txt)
# remove html links
some_txt = gsub("http\\w+", "", some_txt)
# remove unnecessary spaces
some_txt = gsub("[ \\t]{2,}", "", some_txt)
some_txt = gsub("^\\s+|\\s+$", "", some_txt)

# define "tolower error handling" function
try.error = function(x)
{#install.packages("tm.plugin.sentiment", repos="http://R-Forge.R-project.org")
  # create missing value
  y = NA
  # tryCatch error
  try_error = tryCatch(tolower(x), error=function(e) e)
  # if not an error
  if (!inherits(try_error, "error"))
  y = tolower(x)
  # result
  return(y)
}

# lower case using try.error with sapply
some_txt = sapply(some_txt, try.error)

# remove NAs in some_txt
some_txt = some_txt[!is.na(some_txt)]
names(some_txt) = NULL
unq.some_txt <- some_txt[ ! duplicated( some_txt ) ]
print(unq.some_txt)
df<-data.frame(unq.some_txt )
write.xlsx(x = df, file = "lgcamera.excelfile.xlsx",sheetName = "TestSheet", row.names = FALSE)

```

#### AUTHOR PROFILE

**Dr. P.G.Naik**, received M.Sc. degree in Physics and Mathematics and Ph.D. degree in physics from Karnataka University, Dharwad. She received MCA degree from IGNOU with first class Distinction. Currently, she is working as Professor in the Department of Computer Studies, SIBER, Kolhapur. Her areas of interest are network security, soft computing and cloud computing. She has participated in several national and international conferences and has published more than 30 papers in International and national journals of repute.

**Mrs.S.S.Jamsandekar** , received her M.Sc. degree in Computer Science from Pune University and is currently working as Assistant Professor in the department of Computer Studies,Kolhapur. Her areas of interest are soft computing and Data Mining.She has published more than ten papers in International and national journals of repute.

**Mrs.K.S.Mahajan** , received her M.C.A. degree in Computer Application from Bharti Vidyapeeth Pune University and M.Phil from Shivaji University, Kolhapur and is currently working as Assistant Professor in the department of Computer Studies,Kolhapur. Her areas of interest are soft computing and Data Mining.She has published more than six papers in International and national journals of repute.